

PATENT APPLICATION
 Docket No.: CIS00-2411

- 1 -

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as Express Mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on

Date: 6/30/00 Express Mail Label No.: EK408252920US

Signature: J Scott Southworth

Typed or Printed Name: J. Scott Southworth

Inventors:

Lewis D. Dodrill, Ryan A. Danner, and
Steven J. Martin

Attorney's Docket No.: CIS00-2411

APPARATUS AND METHODS FOR PROVIDING AN AUDIBLY CONTROLLED
USER INTERFACE FOR AUDIO-BASED COMMUNICATION DEVICES

5 BACKGROUND

The evolution of the conventional public switched telephone network has resulted in a variety of voice applications and services that can be provided to individual subscribers and business subscribers. Such services include voice messaging systems that enable landline or wireless subscribers to record, playback, and forward voice mail messages. However, the ability to provide enhanced services to subscribers of the public switched telephone network is directly affected by the limitations of the public switched telephone network. In particular, the public switched telephone network operates

5 An open standards-based Internet protocol (IP) network, such as the World Wide Web, the Internet, or a corporate intranet, provides client-server type application services for clients by enabling the clients to request application services from remote servers using standardized protocols, for example, the hypertext transport protocol (HTTP). The web server application environment can include web server software, such as Apache, 10 implemented on a computer system attached to the IP network. Web-based applications are composed of HTML (Hypertext Markup Language) pages, logic, and database functions. In addition, the web server may provide logging and monitoring capabilities.

In contrast to the public switched telephone network, the open standards-based IP network has enabled the proliferation of web based applications written by web application developers using web development tools. Hence, the ever increasing popularity of conventional web applications and web development tools provides substantial resources for application developers to develop robust web applications in a relatively short time and in an economical manner. However, one important distinction between telephony-based applications and web-based applications is that telephony-based applications are state aware, whereas web-based applications are stateless.

In particular, conventional telephony applications are state aware to ensure that prescribed operations between the telephony application servers and the user telephony devices occur in a prescribed sequence. For example, operations such as call processing operations, voicemail operations, call forwarding, etc., require that specific actions occur in a specific sequence to enable the multiple components of the public switched telephone network to complete the prescribed operations.

The prior art web-based applications running in the IP network, however, are state-less and transient in nature, and do not maintain application state because

application state requires an interactive communication between the browser and back-end database servers accessed by the browsers via a HTTP-based web server. However, an HTTP server provides asynchronous execution of HTML applications, where the web applications in response to reception of a specific request in the form of a URL (Uniform Resource Locator) from a client, instantiate a program configured for execution of the specific request, send an HTML web page back to the client, and terminate the program instance that executed the specific request. Storage of application state information in the form of a "cookie" is not practical because some users prefer not to enable cookies on their browser, and because the passing of a large amount of state information as would normally be required for voice-type applications between the browser and the web application would substantially reduce the bandwidth available for the client.

In reference to a conventional telephony-based application (unlike those in the patent applications incorporated by reference above), a user can use the application to access prerecorded responses from a remote source by using audio prompts. This prior art interface may be based on simple predefined voice commands, like "yes" or "no," or reciting a number to select an option. The interface may also be based on entering numbered or other responses on a touch tone keypad into the telephone. For example, a user can use a touch tone telephone to access a bank and obtain the balance or other information on a bank account over a telephone. A user can also use a touch tone telephone to obtain information about some topic or organization they are interested in, such as the hours, exhibits, prices, and special events for a museum, based on a menu of prerecorded prompts and messages maintained by the museum.

In general, in conventional techniques, automated speech recognition techniques (ASR) executing on a processor or computer provide for the recognition of words or phrases in a user's speech. Typically, when sitting at a computer, a user can provide speech input into a microphone attached to a computer, and the computer can translate words and phrases in the speech into commands or data that the computer receives as input similar to the way input typed into a keyboard would be used by a computer. Text



5

10

15

20

25

5

25

5

10

15

25

the user device is capable of interpreting, for example audio signals for an analog telephone, or text for a pager or a facsimile machine.

5
10

20

25

The telephone, or other audio communication device, is not required to have a

microprocessor, specialized electronics, or understand a specific markup language protocol, such as WAP. The audio response or menu is provided from a proxy browser. The proxy browser manages the input from the user, interprets the input with ASR software, and passes on the input and/or request to an audio application program that

5 provides a response or menu that the proxy browser then plays to the user of the telephone. The invention does not require the proxy browser to be a bulky or complex application, and the audio application provides a variety of menus and responses, including user customizable responses and commands, along with access to databases or stored information, such as favored telephone numbers, for the user. Thus the user can

10 hear a personalized menu, including, for example, options to call a favored number often called by the user. Thus the user, can quickly select an option indicating a person to be called. The user can also customize the menu by audibly selecting an option to do so, or entering a command, such as "Add Bob to call list". For example, the audibly controlled interface, according to one embodiment of the invention, can respond by completing the

15 request, if Bob's phone number is already in a database, or query the user for the phone number.

Conventional wireless telephones providing menu services, such as telephones based on a WAP protocol, typically require the telephone to include electronic circuitry or a computer processor programmed to recognize a specialized protocol, such as the WAP

20 protocol. Typically, this type of WAP access is closely integrated with a particular telephone service provider. Conversely, the techniques of the present invention do not require the telephone, or other audio communication device, to be designed or programmed for a specific data protocol, other than the conventional telephony protocols for that type of audio communication device. That is, the techniques of the invention do

25 not require the user's telephone to be designed or programmed for the WAP protocol. This approach allows any type of conventional telephone to use the system of the invention to access, for example, voice mail, electronic mail, sites on the web, etc.

In one embodiment, the invention is directed to a method in a browser for

5
10

15

20

20

25

The method includes, in one embodiment, receiving one or more commands for storing data, retrieving data, and/or placing an outbound telephony call.



15

20

25

In a further embodiment, the speech input information includes one or more commands for storing data, retrieving data, and/or placing an outbound telephony call.

5

15

25



5

10

20

In one embodiment, the invention is directed to a processor-based system for

In one embodiment, the invention is directed to method in an application server. The method includes receiving a first request over a network for a response for a subscriber, accessing profile information for the subscriber from a database, generating a

5

10

15

20

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, features and advantages of the invention will be apparent from the following more particular description of preferred embodiments of the

invention, as illustrated in the accompanying drawings in which like reference characters refer to the same parts throughout the different views. The drawings are not necessarily to scale, emphasis instead being placed upon illustrating the principles of the invention.

Fig. 1 is a block diagram illustrating a paradigm that enables unified voice
5 messaging services and data services to be provided via an IP network using browser audio control according to an embodiment of the present invention.

Fig. 2 is a diagram illustrating in further detail implementation of audio applications on the IP network of Fig. 1 according to an embodiment of the present invention.

Fig. 3 is a diagram illustrating in detail the application server of Fig. 2 according to
10 an embodiment of the present invention.

Figs. 4A and 4B are diagrams illustrating extensible markup language (XML) expressions usable for implementations of voice application on the IP network.

Fig. 5 is a block diagram illustrating in further detail the proxy browser of Fig. 2 according to one embodiment of the invention.

Fig. 6 is a block diagram illustrating an audio interface system including a proxy
15 browser, application server, document database, subscriber database, personalized audio-based menu, and limited communication device according to one embodiment of the invention.

Fig. 7 is a high level flow chart of the process of providing an audio-based
20 response from a proxy browser and application server to a user of a limited communication device according to one embodiment of the invention.

Fig. 8 is a flow chart of the process of providing an audio-based response under the control of a proxy browser, according to one embodiment of the invention.

25 DETAILED DESCRIPTION

The invention is directed to techniques for providing an audibly controlled interface for a user of a limited audio-based communication device, for example, a telephony device such as a desktop telephone or a cellular telephone. The

09608232 "063000

5

10

15

20

25

Figs. 1 through 5 are diagrams illustrating an example of the environment in

Fig. 1 is a block diagram illustrating a unified communications architecture 60

The clients 42a and 42b, referred to herein as “fat clients” and “thin clients”,

respectively, have the distinct advantage that they can initiate requests using IP protocol to any connected web server 64 to execute part or most of the applications 44 on behalf of

the clients. An example of a fat client 42a is an e-mail application on a PC that knows how to run the application 44 and knows how to run the IP protocols to communicate directly with the messaging server via the packet switched network 50. An example of a thin client 42b is a PC that has a web browser 56, which, in this case, can use IP protocols such as HTTP to receive and display web pages generated according to hypertext markup language (HTML) from server locations based on uniform resource locators (URL's) input by the user of the PC.

As shown in Figure 1, each of the clients (tiny clients 18d, 18e, 18f; skinny clients 18a, 18b, 18c; thin clients 42b; and fat clients 42a) are able to communicate via a single, unified architecture 60 that enables voice communications services between different clients, regardless of whether the client actually has browser capabilities. Hence, the fat client 42a and the thin client 42b are able to execute voice enabled web applications without any hardware modification or any modification to the actual browser; rather, the browsers 56 in the clients 42a and 42b merely are provided with an executable voice resource configured for providing browser audio control, described below.

The user devices 18a, 18b, and 18c, illustrated as a cordless telephone 18a, a fax machine 18b having an attached telephone, and an analog telephone 18c, are referred to herein as "skinny clients," defined as devices that are able to interface with a user to provide voice and/or data services (e.g., via a modem) but cannot perform any direct control of the associated access subnetwork.

The wireless user devices 18d, 18e, and 18f, illustrated as a cellular telephone (e.g., AMPS, TDMA, or CDMA) 18d, a handheld computing device (e.g., a 3-Com Palm Computing or Windows CE-based handheld device) 18e, and a pager 18f, are referred to as tiny clients. "Tiny clients" are distinguishable from skinny clients in that the tiny clients tend to have even less functionality in providing input and output interaction with a user, rely exclusively on the executable application in an access subnetwork to initiate communications; in addition, tiny clients may not be able to send or receive audio signals such as voice signals at all.

5

10

20

25

of an HTML page having XML tags for audio control by a voice resource within the browser, or may perform processing and return a calculated value to enable the browser 56 or 62 to perform additional processing.

The application server 66 accesses selected stored XML application pages (i.e., pages that define an application) and in response generate new HTML pages having XML tags during runtime and supply the generated HTML pages having XML tags to the web server 64. Since multiple transactions may occur between the browser 56 or 62 and the application server 66, the application server 66 is configured to store, for each existing user session, a data record, referred to as a "brownie", that identifies the state of the existing user session; hence, the application server 66 can instantiate a procedure, return the necessary data, and terminate the procedure without the necessity of maintaining the instance running throughout the entire user session.

Hence, the application server 66 executes voice application operations from a stored XML document based on a transient application state, where the application server 66 terminates the application instance after outputting the generated XML media information to the browser 62.

Fig. 2 is a diagram that illustrates in further detail the network 60 of Fig. 1, based on Fig. 4 of the above-incorporated application 09/480,485. As shown in Fig. 2, the arrangement of providing browser audio control for voice enabled web applications by the web server 64 and the application server 66 enables voice application services to be implemented in a web server paradigm for many different telephony services, including authentication and billing services 70, domain name services 72, local directory services 74, registry directory and event services 76, and management services 80.

In addition to Fig. 1, Fig. 2 includes PSTN 10, voice resources 86, IP (Internet Protocol) connections 82, routers 84a, 84b, 84c, 84d, IP gateway 87a, 87b, voice over IP interface 88, HTTP connections 89, firewalls 90, gateserver 92, a browser based XML editor tool 94, XML applications and functions 96, dynamic HTML/XML pages 98, and a registry 100. Fig. 2 also illustrates in further detail the browser and web application

09/480,485

5

10

20

25



5

15

20

25

5

15

25

5

10

25

Fig. 4A also illustrates an XML element 110 that defines hotkey pattern values in XML tag 112 that enables the XML aware plug-in resource 86 to initiate execution of a prescribed audio operation in response to detecting a single key input (e.g., *5) by immediately posting the single key input to the server 66 and receiving another XML page that includes the prescribed audio operation, in contrast to HTML form tags, which cannot automatically submit data to a web server upon receipt of a digit. Hence, the ability to define hotkey pattern values enables a web application developer to use XML to define applications that perform voice mail type operations, such as responding to hotkey inputs. As described below, the appearance of hotkey pattern values is implemented merely by posting the input upon receipt thereof, and receiving another XML page.

As described above, the browsers 56 and 62 enable a user's device to be isolated from the web server application side (64, 66) of the network. Hence, user devices (e.g., 18a, 18b, 18c, 42) interacting with the respective browsers 56 or 62 may have completely different hardware or software configurations, as well as different operating environments. In addition, the browser strategy of tagging is flexible, since one user device may not have a microphone, another user device may not have speakers, and another device may not have an alphanumeric or graphical display.

According to the disclosed embodiment, the proxy browser 62 is configured for selectively executing the HTML tags and the XML tags based on capabilities data stored for the corresponding user device 56, 18. In particular, the proxy browser 62 stores for each corresponding user device capabilities data that specifies functions that the user device 56, 18 is able to perform. For example, the capabilities data will specify for each user device 56, 18 whether the user device includes a display for alphanumeric or graphical images, whether the user device includes a sound processor for playing digital audio files via a speaker and recording digital audio files using a microphone, whether the user device has a microphone and speaker for reception and playback of analog audio signals, or whether the user device has a numeric keypad (digital or DTMF) or any key for accepting user inputs.

Fig. 5 is a diagram illustrating in detail the proxy browser 62 configured for providing web application audio control for user devices 18 according to one embodiment of the invention. In Fig. 5, the proxy browser 62 includes a media (voice) browser control 132, a web browser 130, an XML parser 134, and device interface 136. The proxy browser 62 is configured for providing audio control for devices such as an analog telephone 18c via the PSTN 10, or an Internet protocol telephone 120 having an IP connection via an IP-based packet switched network 122, or any of the skinny or tiny client devices 18 as illustrated in Fig. 1. As shown in Fig. 5, the analog telephone 18c may also be connected to the IP network 122 via a voice over IP gateway 124.

The proxy browser 62 includes an HTTP-compliant (i.e. web) browser 130, a voice resource control 132 similar to the voice resource 86 of Fig. 2, an XML parser 134, and a device interface 136. The device interface 136 provides a connection for the proxy browser 62 with the user device, for example the telephone 18c or the IP telephone 120, via the corresponding user device access network 10 or 122. In particular, the device interface 136 includes network specific hardware interface cards and associated drivers that enable the voice resource control 132 to send voice audio commands to the device interface 136; the device 136 then implements those audio commands according to the specific protocols of the network coupling the user device to the proxy browser 62.

For example, the device interface 136 includes an IP network interface card 138 that includes an Ethernet (IEEE 802.3) network interface card 140, and voice over IP control software 142 that is compliant, for example, with Recommendation H.323 from the Telecommunication Sector of the International Telecommunication Union (ITU-T)). The control software 142 serves as the driver software for the Ethernet card 140. The voice over IP control software 142 may include a set of application programming interfaces 144 that enable the voice resource control 132 to issue function calls to the voice over IP control software 142; alternatively, the voice resource control 132 may issue function calls directly to the voice over IP software 142, bypassing the API routines 144. Exemplary H.323 compliant IP network interface cards 138 are available from

10

15

20

25

conventional telephony or other two-way communication protocols for audio devices, but typically is limited in its ability to understand and process other protocols. For example, the limited communication device 304 is not designed to process Internet or WAP protocols, and does not include a web browser allowing direct access to the web without some intermediary device. In an additional example, the limited communication device does not include software or hardware to process markup languages, such as HTML, XML, or WML.

The network browser or proxy browser 306 is an alternate embodiment of the proxy browser 62 of Fig. 2 and provides access to a network, such as the Internet (e.g. 50, as shown in Fig. 1). The proxy browser 306 includes an ASR module 310, device interface 312, and a web browser 340 in communication with the web server 64 over a network. In one embodiment, the ASR module 310 is a software module executing on the proxy browser 306. The ASR module 310 translates audio signals, such as speech information, into words or phrases in a data or text format that the proxy browser 306 can process. The web browser 340 is an alternate embodiment of the web browser 130 of Fig. 5. The device interface or interface connection 312 is a communication interface providing a connection to the limited communication device 304. The device interface 312 is one example of the device interface 136 illustrated in Fig. 5.

The application server 66 is in communication with the web server 64. The application server 66 includes, in an embodiment of the present invention, the call services application 318. The call services application 318 is an executable resource 318 on the application server 66. The call services application 318 includes scripts, procedures and other software entities, such as procedures 228, as shown in Fig. 3, and one or more application-defining tagged documents 328 (e.g. XML menu/decision documents) stored in an application document database 96 and executed in the application runtime 224 of the application server 66. The call services application 318 initiates a call service, such as an outgoing telephony call, based on a verbal command, such as "Call Bob." In addition to placing an outgoing telephony, call services can

include accessing voice mail, accessing electronic mail, accessing a telephone number or other data from a database, adding a telephone number and name, or other data, to a database, or providing other services, based on the user's verbal request.

The application server 66 is in communication with a user database 350 that includes profile information for subscribers to the call services provided by the call services application 318. The call services application 318 uses the subscriber database 350 to obtain data needed to interpret the subscriber's commands, such as Bob's telephone number, when the subscriber makes a verbal request to call Bob. In other embodiments, the database 350 is an LDAP directory storing a profile for the subscriber, or an IMAP database storing a directory for the subscriber. The subscriber can supply the personalized data when initially subscribing to the call services application 318 (e.g. by filing out a hardcopy or electronic form), by adding a name and phone number using a verbal command, providing the data by electronic mail, or by other approaches commonly used to supply data to a database.

15 In one embodiment, a computer program product 380 including a computer readable medium (e.g. one or more CDROM's, diskettes, tapes, etc.) provides software instruction for the proxy browser 306 and/or the call services application 318. The computer program product 380 can be installed by any suitable software installation procedure, as is well known in the art. In another embodiment, the software instructions
20 for the proxy browser 306 and/or the call services application 318 can also be downloaded over a wireless connection. A computer program propagated signal product 382 embodied on a propagated signal on a propagation medium (e.g. a radio wave, an infrared wave, a laser wave, sound wave, or an electrical wave propagated over the Internet or other network) provides software instructions for the proxy browser 306
25 and/or the call services application 318. In alternate embodiments, the propagated signal is an analog carrier wave or a digital signal carried on the propagated medium. For example, the propagated signal can be a digitized signal propagated over the Internet or other network. In one embodiment, the propagated signal is a signal that is transmitted

over the propagation medium over a period of time, such as the instructions for a software application sent in packets over a network over a period of seconds, minutes, or longer.

Fig. 7 is a high level flow chart of the process of providing an audio-based response from a proxy browser 306 and application server 66 to a user of the limited communication device 304 according to one embodiment of the invention. In step 400, the user of a limited communication device 304 provides input, generally referred to as input information 321, by accessing the device 304 or speaking a command into it. Alternately, the user can provide a DMTF (discrete multitone frequency) input by pressing one or more keys on the keypad of the limited communication device 304. In step 402, the device interface 312 receives the input information 321 from the limited communication device 304. For example, the device interface 312 receives input information 321 from an analog telephone 18c via a PSTN 10 or from an IP telephone 120 having an IP connection via an IP-based packet network 122 (as shown in Fig. 5). In one embodiment, the input information 321 is an initial access input 321-1, and the device interface 312 receives the initial access input 321-1 indicating an initial access to the limited communication device, such as detecting an "off hook" condition for an analog telephone 18c. In one embodiment, the proxy browser 306 is capable of connection to several limited communication devices 304 at one time. For example, the proxy browser 306 can watch or monitor the limited communication devices 304. When a user picks up the handset of the telephone 18c, the device interface 312 receives the input 321-1 through a PSTN gateway 10 (alternatively, through a IP gateway 122 or other gateway). The proxy browser 306 identifies the identity of the limited communication device 304 when it is accessed. For example the proxy browser 306 identifies the phone number of the limited communication device 304. If the user places a call, the proxy browser 306 identifies the originating telephone number (e.g. the telephone number of the limited communication device 304), the called number, and the forwarding telephone number (if applicable). (See Fig. 8 for a more detailed flowchart of a sample process followed by the proxy browser 306 for receiving input from the user.)

The web server 64 receives the request URL request 326 (as discussed above) that represents the input information 321 from the web browser 340 associated with the proxy browser 306 (step 406). The application server 66 then prepares a response 330 to the URL request 326 based on one or more application-defining documents 328 stored in the document database 96. In one embodiment, the application server 66 accesses profile information from a subscriber database 350 that contains profiles of subscribers to the services provided by the call services application 318. The database 350 includes personalized data for each subscriber. For example, application server 66 uses the database 350 to determine what telephone number to call when the subscriber verbally requests that a telephone call be made to some individual.

5 application server 66 and the application-defining documents 328. The application server 66 generates a menu-based or other response suitable for audio output based on the application-defining document 328 and the URL request 326 representing the input information 321 (step 410). For example, if the input information 321 is an input 321-1 indicating an initial access to the limited communication device 304, then the application
0 server 66 uses a TTS technique to convert the text information in an XML document 328 into an audio initial menu in the form of audio files (such as .wav files).

15 the device interface 312 to the limited communication device 304 (step 412).

device 304 as the response to the user picking up the handset. The user can then provide the speech input information 321-2 by selecting one of the options of the menu 308 either by saying the number, or saying a verbal command from a menu (e.g. the personalized menu 308), such as “Call Bob.” Alternatively, the audio output provided by the application server 66 and played by the proxy browser 306 is a welcome tone (i.e. a tone different from the conventional dial tone). After hearing the welcome tone, the user can speak verbal commands such as “Hello,” “Call,” or “Messages.” Then the user provides additional speech input information 321-2, which is received by the proxy browser 306 (step 402), and another response is generated (steps 404 through 412). This speech input

Fig. 8 is a flow chart of the process of providing an audio-based response under the control of a proxy browser 306, according to one embodiment of the invention. First, the proxy browser 306 detects an input 321 from the limited communication device 304, which is speech input information 321-2, or other input, such as an input 321-1 indicating initial access to the device 304, as described above. The proxy browser 306 establishes an active session with the limited communication device 304 (step 500), and the web browser 340 provides an initial URL request 326-1 to the application server 66 (step 502), as described above. The web server 64, in response to receiving the request 326-1, forwards the request 326-1 to the application server 66 across a connection, for example, a common gateway interface (CGI). The application server 66 returns a response document 330 in the form of a generated page, such as a page including HTML and XML tags, to the proxy browser 306.

The proxy browser 306 then checks in step 506 whether the XML data includes control data that is essential to provide voice application control to the limited communication device 304. If the proxy browser 306 determines in step 506 that the response document 330 does not include essential control data from the application server 66, then the proxy browser 306 plays a stored system level prompt in step 508 indicating that service is unavailable.

If the proxy browser 306 determines in step 506 that the response document 330 includes the essential control data, then, in step 510, the proxy browser 306 begins to play

5

14

25

524). If no record operation is pending, then the proxy browser 306 posts the matched key chunk data to the URL for the application server 66 providing the call services application 318. Hence, the proxy browser 306 can provide the appearance to the user that speaking an option from the menu 308 causes an immediate response in the proxy browser 306 to play another audio file, when actually the web browser 340 merely posts the user's matched key chunk data in the URL request 326-2 to the call services application 318, causing the application 318 to return another response document 330 including the sound to be played.

If a record operation is pending, the proxy browser 306 begins recording (step 526), for example by playing a tone to signal the user to begin speaking. In another embodiment, the proxy browser 306 records whatever the user provides as input 321-2 after the playing of an audio file, without a specific tone. The proxy browser 306 continues to record until a time-out occurs (e.g., 30 seconds), or the user enters a key or voice command indicating recording should be halted. In another embodiment, the ASR module 310 processes the input 321-2, determines when a key chunk match has occurred, and then halts the recording.

While this invention has been particularly shown and described with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the invention as defined by the appended claims.

For example, the application server 66 can use an ASR technique to process the input information 321 received from the proxy browser 306 without any ASR processing in the proxy browser 306. In general, either one of the ASR and TTS techniques can be performed in either the proxy browser 306 or the application server 66. For example, the proxy browser 306 can perform the TTS technique to translate generated text output provided by the application server 66 into an audio file.

In addition, the proxy browser 306 and application server 66 are not required to be connected by the Internet, but may be connected by other types of network or direct line

connections, as is known in the art. Also, the functions and capabilities of the proxy browser 306 and application server 66, as described herein, can be implemented on one computer system, rather than separate computer systems, or on many computer systems, such as in a distributed object or other distributed computing approach.